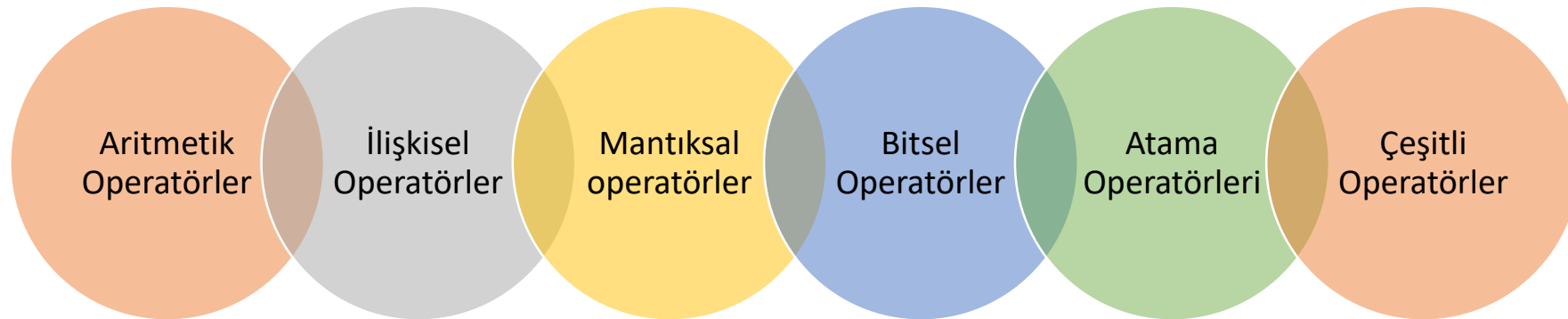


Operatörler

Operatörler

- Operatör, derleyiciye belirli matematiksel veya mantıksal işlemleri gerçekleştirmesini söyleyen bir semboldür.
- C# zengin yerleşik işleçlere sahiptir.

Operatörler



Aritmetik Operatörler

+	İki işlenen ekler	$A + B = 30$
----------	--------------------------	--------------------------------

-	İlk işlenenden ikinci işleneni çıkarır.	$A - B = -10$
----------	--	---------------------------------

*	Her iki işleneni de çarpar.	$A * B = 200$
----------	------------------------------------	---------------------------------

/	Payı, numaratörden ayırıcıya böler.	$B / A = 2$
----------	--	-------------------------------

Aritmetik Operatörler

%	Modül İşleci ve bir tamsayı bölmesinden sonra kalanı	B% A = 0
++	Arttırma operatörü, tamsayı değerini bir artırır	A ++ = 11
--	Azaltma operatörü tamsayı değerini bir azaltır	A-- = 9

İlişkisel Operatörler

Operatör	Açıklama	Örnek
==	İki işlenenin değerlerinin eşit olup olmadığını kontrol eder, evet ise koşul doğru olur.	(A == B) doğru değil.
!=	İki işlenenin değerlerinin eşit olup olmadığını kontrol eder, değerler eşit değilse koşul doğru olur.	(A != B) doğrudur.
>	Soldaki işlenenin değerinin sağ işlenenin değerinden büyük olup olmadığını kontrol eder, evet ise koşul doğru olur.	(A > B) doğru değil.
<	Soldaki işlenenin değerinin sağ işlenenin değerinden küçük olup olmadığını kontrol eder, evet ise koşul doğru olur.	(A < B) doğrudur.

İlişkisel Operatörler

Operatör	Açıklama	Örnek
\geq	Sol işlenenin değerinin sağ işlenenin değerinden büyük veya ona eşit olup olmadığını kontrol eder, evet ise koşul doğru olur.	$(A \geq B)$ doğru değil.
\leq	Sol işlenenin değerinin sağ işlenenin değerinden küçük veya ona eşit olup olmadığını kontrol eder, evet ise koşul doğru olur.	$(A \leq B)$ doğrudur.

Mantıksal Operatörler

Operatör	Açıklama	Örnek
&&	Mantıksal AND operatörü çağrıldı. Her iki işlenen de sıfır değilse, koşul doğru olur.	(A && B) yanlıştır.
	Mantıksal VEYA Operatörü çağrıldı. İki işlenenden herhangi biri sıfır değilse, koşul doğru olur.	(A B) doğrudur.
!	Mantıksal NOT Operatörü olarak adlandırıldı. İşleneninin mantıksal durumunu tersine çevirmek için kullanılır. Bir koşul doğruysa, Mantıksal NOT operatörü yanlış yapar.	! (A && B) doğrudur.

Bitisel Operatörler

p	q	p & q	n m	p q
0	0	0	0	0
0	1	0	1	1
1	1	1	1	0
1	0	0	1	1

Bitset Operatörler

A = 60

0	0	1	1	1	1	0	0
---	---	---	---	---	---	---	---

B = 13

0	0	0	0	1	1	0	1
---	---	---	---	---	---	---	---

A&B

0	0	0	0	1	1	0	0
---	---	---	---	---	---	---	---

Bitisel Operatörler

A = 60

0	0	1	1	1	1	0	0
---	---	---	---	---	---	---	---

B = 13

0	0	0	0	1	1	0	1
---	---	---	---	---	---	---	---

A|B

0	0	1	1	1	1	0	1
---	---	---	---	---	---	---	---

Bitset Operatörler

A = 60

0	0	1	1	1	1	0	0
---	---	---	---	---	---	---	---

B = 13

0	0	0	0	1	1	0	1
---	---	---	---	---	---	---	---

$A \wedge B$

0	0	1	1	0	0	0	1
---	---	---	---	---	---	---	---

Bitsel Operatörler

A = 60

0	0	1	1	1	1	0	0
---	---	---	---	---	---	---	---

$\sim A$

1	1	0	0	0	0	1	1
---	---	---	---	---	---	---	---

Bitsel Operatörler

A = 60

0	0	1	1	1	1	0	0
---	---	---	---	---	---	---	---

$\sim A$

1	1	0	0	0	0	1	1
---	---	---	---	---	---	---	---

Bitsel Operatörler

Operatör	Açıklama	Örnek
&	İkili VE İşleci, her iki işlenende de varsa sonuca bir bit kopyalar.	$(A \& B) = 12, 0000\ 1100$
	İkili VEYA İşleci, işlenenlerden herhangi birinde varsa bir bit kopyalar.	$(A B) = 61, 0011\ 1101$
^	İkili XOR Operatörü, bir işlenen içinde ayarlanmışsa, ancak her ikisinde birden değil ise biti kopyalar.	$(A \wedge B) = 49, 0011\ 0001$

Bitsel Operatörler

Operatör	Açıklama	Örnek
~	İkili Birler Tamamlayıcı Operatörü tekli ve bitleri 'çevirme' etkisine sahiptir.	(~ A) = -61, işaretli bir ikili sayı nedeniyle 2'nin tamamlayıcısında 1100 0011'dir.
<<	İkili Sol Kaydırma Operatörü. Soldaki işlenenler değeri, sağ işlenen tarafından belirtilen bit sayısı kadar sola taşınır.	A << 2 = 240, yani 1111 0000
>>	İkili Sağ Shift Operatörü. Soldaki işlenenler değeri, sağ işlenen tarafından belirtilen bit sayısı kadar sağa taşınır.	A >> 2 = 15, 0000 1111

Atama Operatörleri

Operatör	Açıklama	Örnek
=	Basit atama operatörü, sağ taraftaki işlenenlerden sol taraftaki işlenenlere değerler atar	$C = A + B$, $A + B$ 'nin değerini C 'ye atar
+=	VE atama operatörü ekle, Sol operanda sağ operandı ekler ve sonucu sol operanda atar.	$C += A$, $C = C + A$ 'ya eşdeğerdir
-=	Çıkar VE atama operatörü, Sağ operandı sol operanddan çıkarır ve sonucu sol operanda atar.	$C -= A$, $C = C - A$ 'ya eşdeğerdir
*=	AND atama operatörünü çarpın, sağ operandı sol operandla çarpar ve sonucu sol operanda atar	$C * = A$, $C = C * A$ 'ya eşdeğerdir

Atama Operatörleri

Operatör	Açıklama	Örnek
/=	VE atama operatörünü böl, Sol operandı sağ operandla böler ve sonucu sol operanda atar.	$C / = A$, $C = C / A$ 'ya eşdeğerdir
%=	Modül VE atama operatörü, İki işlenen kullanarak modülü alır ve sonucu sol operanda atar	$C \% = A$ eşdeğerdir $C = C \% A$
<<=	Sol kaydırma VE atama operatörü	$C << = 2$, $C = C << 2$ ile aynıdır
>>=	Sağ kaydırma VE atama operatörü	$C >> = 2$, $C = C >> 2$ ile aynıdır

Atama Operatörleri

Operatör	Açıklama	Örnek
& =	Bitsel AND atama operatörü	$C \&= 2$, $C = C \& 2$ ile aynıdır
^=	bit düzeyinde özel OR ve atama operatörü	$C \wedge= 2$, $C = C \wedge 2$ ile aynıdır
 =	bitsel kapsayıcı OR ve atama operatörü	$C \mid= 2$, $C = C \mid$ ile aynıdır $\mid 2$

Çeşitli Operatörleri

Operatör	Açıklama	Örnek
sizeof()	Verinin boyutunu döner.	sizeof(int), returns 4.
typeof()	Bir sınıf türünü döndürür.	typeof(StreamReader);
&	Bir değişkenin adresini verir.	&a; returns actual address of the variable.

Çeşitli Operatörleri

Operatör	Açıklama	Örnek
*	Bir değişkene işaretçi.	İşaretçi tanımlar.
? :	Koşul ifadesi.	3 bir koşul ifadesidir.
is	Bir nesnenin belirli bir türde olup olmadığını belirler.	Tür kontrolü gerçekleştirir.
as	Cast işleminin başarısız olması durumunda bir istisna oluşturmaz.	

Operatör Önceliği

Kategori	Operatör	İlişkisellik
Postfix	() [] -> . ++ --	Left to right
Unary	+ - ! ~ ++ -- (type)* & sizeof	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right

Operatör Önceliği

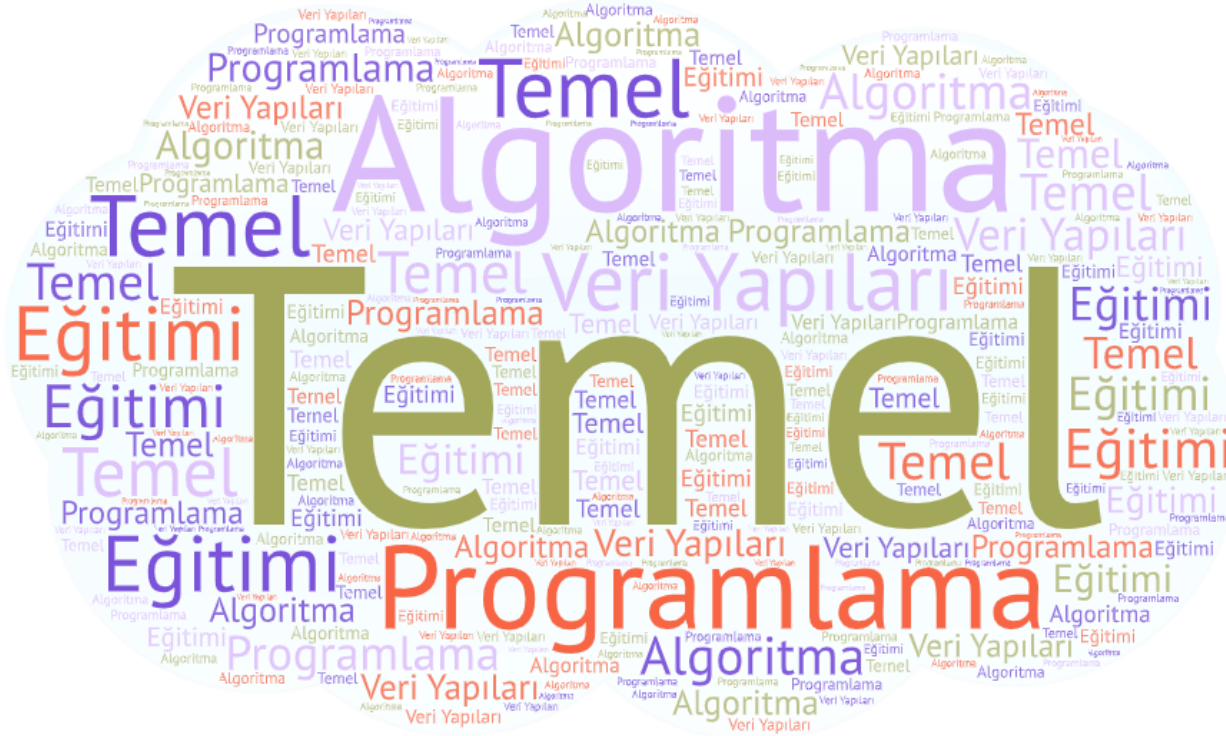
Kategori	Operatör	İlişkisellik
Shift	<< >>	Left to right
Relational	< <= > >=	Left to right
Equality	== !=	Left to right
Bitwise AND	&	Left to right

Operatör Önceliği

Kategori	Operatör	İlişkisellik
Bitwise XOR	<code>^</code>	Left to right
Bitwise OR	<code> </code>	Left to right
Logical AND	<code>&&</code>	Left to right
Logical OR	<code> </code>	Left to right

Operatör Önceliği

Kategori	Operatör	İlişkisellik
Conditional	?:	Right to left
Assignment	= += -= *= /= %= >>= <<= &= ^= =	Right to left
Comma	,	Left to right



Operatörler